

IN THE SPECIFICATION

Please amend the specification as follows:

(Page 6, lines 7-17) An illustrative embodiment of the present invention utilizes a parsing utility with a generic file parser 30 at connection database 20 to define the most common way to parse a transaction data file. The generic parser defines the typical sequence of events happening during the parsing process. Also it defines the typical data identification and data extraction. The generic parser can typically handle a majority of transaction data. For transaction data in other formats, a specific parser 32a, 32b, or 32c can overwrite any of the functions provided by the generic parser 30. For example, company identification function, card identification function, currency identification function, transaction type mapping function, etc. within the generic parser 30 could all be overwritten by a specific parser. With a minimum effort for specific data parsing 32a, the format 24 from the data provider 22a can be parsed. This allows the illustrative embodiment to easily process a majority of file formats such as format 24.

(Page 8, lines 7-24) The next step of preliminary processing of transaction data is shown in Fig. 3A. The transaction data now in the common data format 34 is processed for insertion into the database system 42. The transaction data in the common data format 34 is processed further at data import processing 36. This data processing can provide automatic cost allocation 38 or tax recordation 40 (shown in Fig 1). The transaction data is first processed by a loader processor 50 that loads the transaction data in to a set of temporary tables 52 (shown in Fig. 4). The loader processor 50 is typically an SQL loader processor, thereby sorting the transaction data in accordance with the programmed queries to sort the data into the appropriate temporary tables 52. The six temporary tables used by the illustrative embodiment include transactions 254, line items 56, additional data 58, enhanced data 60, trip leg data 62 and card balance information 64. More information on these tables is shown in Fig. 4. The pre-processing of the data into these temporary tables 52 allows the system to then do a sort of the data 66 Fig. 3A, for the proper format for ultimate entry into the database 42. Since the structure and format of the data for the database 42 is very different than the input format of transaction line feed data from the card providers, this use of temporary tables 52 is an intermediate step that assists in sorting the data before it is ready to be placed in the database 42. Temporary tables typically do not have interdependencies or linked relations, which happens later in the process of data sorting

discussed hereinafter. The temporary tables also hold more information than is ultimately stored in the database 42. This information is useful for data processing operation 68 and operation 70, for example extra columns in the tables for record types, and flags.

(Page 11, line 27 – Page 12, line 4) The card balance information for the transaction is finally written to the database, step 142. Again, if an error occurs, it is recorded to the error table, step 136. The system then repeats to read the next entry step 128 and repeat the cycle. When the end of file is detected, step 130, the access to the temporary files is closed, step 144. At step 146, the system makes a list of all new cards set up as part of step 134. This list is then used to indicate the number of new cards detected and the system can then be updated with the missing information for those new cards. At step 148, the contents of the card balance table are deleted so that the records are not encountered again in the case of a restart.

(Page 13, lines 10-20) The next step for the processing of transactions as outlined in Fig. 5 is the processing of trip leg data 76, with details provided in Fig. 10. Trip legs are generally related to information regarding travel charges, typically flight or train information, etc. This information is valuable for travel expense claim by providing details on the nature of the expenses. The trip leg temporary table 62 Fig. 3A is accessed, step 194 Fig. 10. The next item is read, step 196, and if it is for a card that has expired, the card is reactivated as previously described, step 138. If any errors occur, they are recorded to the trip leg error table, step 192. Otherwise an airport terminal identification is linked in for the record, step 200, and the trip leg record is written into the database, step 202. This processing continues with the following entries in the temporary table until all entries have been read, step 198. The temporary table is then closed, step 204, and it is then deleted, step 206.